MODELING OF COGNITIVE CLASSIFICATION AND SEGMENTATION OPERATIONS DURING MUSIC LISTENING WITH MORFOS MUSIC ANALYSIS SOFTWARE

Joséphine Calandra[abc]
josephine.calandra@ircam.fr

Jean-Marc Chouvel[a]
jeanmarc.chouvel@free.fr

Myriam Desainte-Catherine[c]
myriam@labri.fr

[a]Sorbonne Université, IReMus, UMR 8223, Paris, France;
[b]Sorbonne Université, Collegium Musicæ, IRCAM -STMS, UMR 9912, Paris, France;
[c]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France

ABSTRACT

This article is part of an interdisciplinary exploration in computational musicology and mathematical theory. It presents MORFOS, a music analysis software that aims to model cognitive phenomena while listening to music. This model is based on two operations: classification and segmentation, which are operated at different time scales simultaneously. These operations, applied either to audio or symbolic music, lead to the creation of a musical form's representation that is called a multi-scaled formal diagram. This representation corresponds to a high-level notation that helps musicians better understand musical structure. As the software models a cognitive process that happens alongside the musical flow, the software also works in real-time alongside its acquisition. In this article, we present the formalisation and the implementation of classification and segmentation of the algorithm implemented in MORFOS, called the Cognitive Algorithm.Then, we present a first analysis of the software's behaviour depending on the choices made by the algorithm during these operations, defining distinct Cognitive Phases. We then present four analyses based on four segmentations of the Menuet K.545's *Rondo* of W.A. Mozart and study the computed Cognitive Phases, in order to better understand how music's structure impacts cognitive mechanisms while listening to music.

1. INTRODUCTION

In this article, we aim to gain a better understanding of the cognitive phenomena associated with musical practice, and more specifically with music listening. To this end, we present a multi-scale music analysis software that aims to model real-time classification and segmentation operations taking place simultaneously at distinct temporal scales. The result of the multi-scaled segmentation and classification processes on audio or symbolic music in MORFOS is a representation of the cognitive organisation of musical material over time, called musical form, which is represented in multi-scale formal diagrams created while the musical material is processed in the system.

1.1 Multi-scale music analysis software

The human brain is capable of structuring time at different scales: when we listen to a musical work, we can propose a segmentation both on the scale of the pattern and on the scale of the musical section [1].

A fundamental contribution in the representation of the multi-scale form is Lerdahl and Jackendoff's Generative Theory of Tonal Music [2]. It has been a real breakthrough in music structure analysis, which has been automated by Tojo et al. [3]. Nevertheless, their scope is limited to tonal music. Thus, the Cognitive Algorithm [4] and the MORFOS software [5] represent important advancements in the field of structure and form analysis, as they aim at analysing any kind of music genre, as the cognitive system might do. A parametrisation system influences the segmentation and classification behaviours of the software, allowing the user's choice to simulate a subjective behaviour. Segmentation and classification operations are made at different time scales, starting with elements of the smallest analysis window size, then aggregating the elements to create higher-level elements based on the parameterization.

1.2 Segmentation and classification

N. Meeùs [6] presents segmentation as a ”fundamental activity of musical analysis”, intrinsic to the existence of musical form. He describes the segmentation of a musical work as its ”resolution into units distributed along the time axis, with duration being an essential characteristic.” Meeùs compares musical segmentation to linguistic segmentation, highlighting their similar essential nature. While linguistic segmentation is possible due to the linear nature of language, which is also the case for music, the linguistic discourse — and even more so, the musical discourse — can be multidimensional and contrapuntal. For this paper, in the context of MORFOS, we set aside the complex issue of counterpoint and source separation, as these topics require specific in-depth study.

In the listening model presented by E. Bigand [7], segmentation during musical listening occurs after signal processing based on grouping principles derived from Gestalt theory. Music information retrieval addresses segmentation in order to estimate the overall structure of a musical work, and determine tempo and rhythm [8, 9], which can help identify musical metrics and structures.

Similarity between musical pieces or within a single piece makes it possible to classify works by style or recognize a music group from the very first seconds of listening thanks to specific timbres (Samson [10]).

In the field of Music Information Retrieval (MIR), classification is addressed using various criteria: Gabbolini et al. [11] introduce a similarity measure calculated from a measure of "interest" of the paths between the representations of the two entities in a graph in order to classify genre and artists. Herremans et al. [12] propose statistical models and decision trees to distinguish composers such as Bach, Haydn, and Beethoven. Sarfati et al. [13] use a clustering method to create a graph of relationships between audio files for cover detection. Dehaas et al. [14] implement a local alignment algorithm for chord recognition.

As segmentation seems to be a major process in music listening, it constitutes the first principal operation in MORFOS software. Even though some of the corresponding computer techniques obtain state-of-the-art results by using neural networks, their use presents issues related to explainability. Additionally, some clustering methods require complete prior knowledge. Consequently, MORFOS opts to avoid these techniques for segmentation and classification. Instead, the software is rule-based, offering a range of descriptors (signal-based) and grammar rules (symbolic-based) chosen by the user. These rules might be selected according to the specification of the listening model (timbral or pitch-based, distinguishing more or less the elements...) or the analytical specificities needed for such or such musical genre.

Classification is then the second principal operation in MORFOS Software. For now, even though, from a cognitive point of view, two occurrences of the same note played at two different times can be considered different dissimilar, and the same piece played by two distinct artists may be considered as two different versions, MORFOS works with a binary concept of similarity that depends on the parameters chosen by the user (choice of a similarity threshold). But this can be challenged by displaying a "similarity value" of the current object to other objects as in a Self-similarity matrix (Foote [15]).

1.3 Attention phenomenon modeling

Inspired by L. Meyer's application of the laws of the *Gestalt*, E. Narmour presents in [16, 17] the *Implication-Realisation* model. This model aims to present the mechanisms of attention based on its cognitive aspects during musical listening. Two statements concerning this model are, on the one hand, the realisation or non-realisation of the following two hypotheses: $(1)A + A \rightarrow A$ and, on the other hand, $(2)A + B \rightarrow C$ where A, B, and C are musical groupings. Assumption (1) means that the repetition of the same musical motif implies a successive repetition of that motif, while assumption (2) means that the succession of two distinct musical motifs implies a third distinct motif. The second statement defines whether or not the previously described patterns imply a closure of the musical phrase. The *System & Contrast* [18] model from F. Bimbot et al. describes the internal organisation and relationships between the properties of musical elements grouped into squares of four elements.

J.M. Chouvel presents in [4] an analysis of the behaviour of the Cognitive Algorithm, giving rise to four cognitive phases at each level, corresponding to the four possibilities of results for the two tests of classification and segmentation. It is this model that we will explain in the next sections and put into perspective with the software results.

In the following part, we present briefly the MORFOS Software. Section 3 describes the out-of-time and real-time mathematical formalisation of multi-scale formal diagrams relying on segmentation and classification operations, and Section 4 explains the implementation of the segmentation and classification algorithms. Section 5 presents the meta-analysis of MORFOS' Cognitive Algorithm with the implementation of the cognitive phases that are based on classification and segmentation behaviours, in order to better understand the cognitive model.

2. THE MORFOS MUSIC ANALYSIS SOFTWARE

Analysing a musical flow as "listened to" implies proceeding to (at least) three operations: segmentation, classification, and hierarchisation. The goal of MORFOS is to model those operations in order to represent music strategies for listening and the phenomenon of attention, i.e., for example, surprise or memorisation.

2.1 The Cognitive Algorithm

MORFOS analysis is based on the Cognitive Algorithm, which is documented in [4]. The Cognitive Algorithm was proposed as an analytic methodology by Jean-Marc Chouvel and comes from cognitive musicology [19] and semiology [20, 21].

The Cognitive Algorithm comprises two main steps. The first one is the *Paradigmatic Recognition Test* (classification test). In this test, the musical fragment currently acquired in the algorithm (at first, the analysis of the musical element of the size of the analysis windows for signal, or the parsing of the first annotation for symbolic representation) is compared with previously known fragments of the same level and classified according to this comparison (the first fragment is by default considered as a new element, as the methodology asserts that no prior knowledge is necessary at this stage).

Next, a second test follows: the *Syntagmatic Recognition Test* (segmentation test) determines whether or not the group of fragments that have just been heard constitutes a fragment of the upper level. If this is the case, both tests are repeated at the higher level, a higher level being a level in which the fragments are longer, as they are the result of a concatenation of fragments from the lower level. Thus, each level leads to the realisation of a specific diagram called *formal diagram.*

This algorithm is used as a process for musicologists and had never been implemented. The MORFOS software is based on the concepts provided by this analytic method, while adapting to operational needs. For example, the operations of classification and segmentation have been reversed, as the first operation in the system is the segmentation of the audio stream or the parsing of the symbolic stream.

2.2 Formal Diagram and Multi-scale Formal Diagrams

A Formal Diagram (or Paradigmatic Diagram) is the representation of *musical materials* according to *time* and can be represented simultaneously at different time scales. We show an formal diagram obtained automatically with MORFOS at a two-bar musical scale of Wolfgang Amadeus Mozart Menuet's *Rondo K.545* in Figure 1.

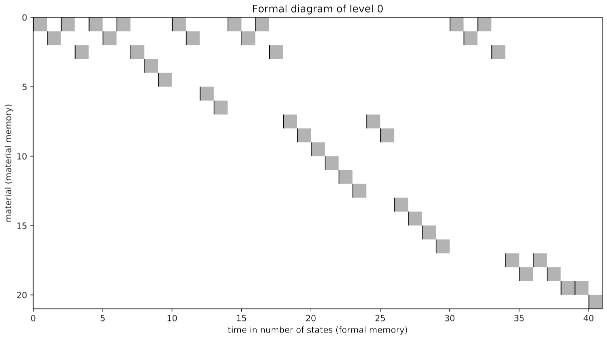

Figure 1. Automated Formal Diagram obtained with MORFOS of W.A. Mozart Menuet's *Rondo K.545* at two bars musical scale, from a symbolic representation.

Materials are defined as classes of substitutable elements. For instance, the concept of the note $A3$ represents a material, but materials can encompass various types, such as musical phrases in some musical styles, or even noises. This concept is closely related to the idea of paradigm. An *object* then refers to a specific occurrence of a material in a musical work.

Starting with the smallest time scale diagrams, the Cognitive Algorithm constructs larger time scale diagrams, so that the sequences of lower-level objects constitute a single higher-level object. For example, a sequence of notes in a formal diagram constitutes a phrase for the higher-scale formal diagram, and a sequence of phrases constitutes a part of the higher-scale formal diagram.

The superimposition of Formal Diagrams obtained at different time scales is called *Multi-scale formal diagram.* A representation of W.A. Mozart's *Rondo K.545* Multi-scale Formal diagram is shown in Figure 2 and more deeply explained in [22].

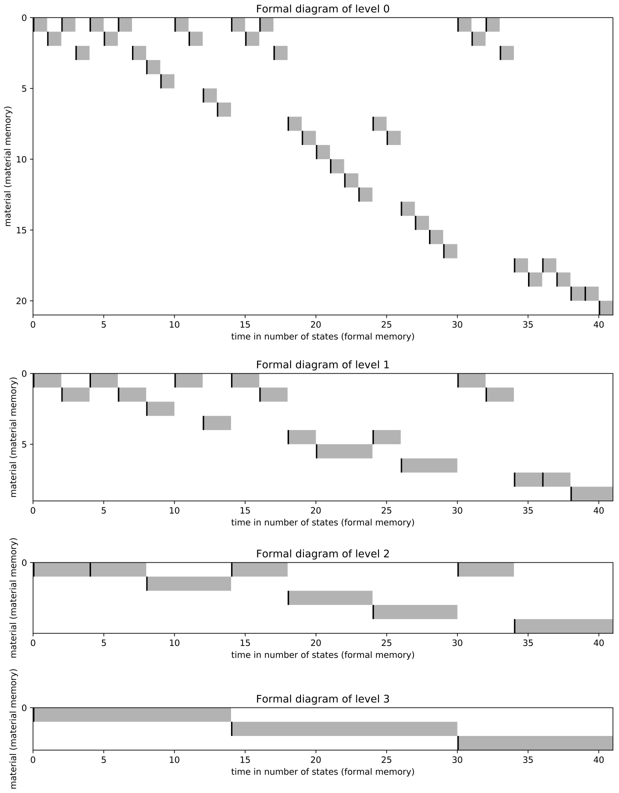

Figure 2. Automated Multi-scale formal diagram of W.A. Mozart's *Rondo K545* obtained with MORFOS. Level 0 is at (about) two bars scale, Level 1 is at four bars scale, Level 2 is at height bar scale, and Level 3 is at the sixteen bars scale. Each scale is automatically built according to the construction of the lower one.

3. FORMALISATION OF THE MULTI-SCALE FORMAL DIAGRAM

This section describes the out-of-time and real-time formalisation of the multi-scale formal diagrams in MORFOS.

3.1 Out-of-time formalisation

Let us first note that the input and output at a given level of the MORFOS Algorithm are actually the same data structure. The input of an instance of the MORFOS algorithm can thus be the output of a lower-level instance.

We thus define $r^{(n)}$ as the *representation of level n* equivalent to the *formal diagram of level n* as a word

$$r^{(n)} = x_1^{(n)}...x_i^{(n)}...x_{l_n}^{(n)}$$

with $x_i^{(n)}$ letters defined on the ordered alphabet $\mathcal{A}^{(n)}$.

The symbolic representation is then the succession of characters corresponding to the objects (that can be either symbolic or audio data) represented by the labels of the associated initial materials. These labels have no semantic interest. The *index* of a labeled object in a word $r^{(n)}$ is also defined as the number of the object's location in the word r: it is thus indexed by its index. The notation $x_i = r[i]$ is then accepted.

The level n representation is segmented into a sequence of fragments. The segmentation operation $sgm(r^{(n)}, cs(n))$ takes as arguments the initial representation $r^{(n)}$ and a set of segmentation criteria cs depending of the level n. In fact, the segmentation criteria can vary from one time scale to another and therefore from one level to another. Segmentation criteria may include, for example, a change in a sequence of similar elements at the level of a few milliseconds and the return to the beginning of a sequence in higher levels. Let $s^{(n)}$ be a *segmented representation of level n* which is the grouping of letters from $\mathcal{A}^{(n)}$ into factors $w_j^{(n)}$ belonging to $\mathcal{A}^{(n)*}$ by associativity of the monoid $\mathcal{A}^{(n)}$ on $\mathcal{A}^{(n)*}$ such that

$$s^{(n)} = sgm(r^{(n)}, cs(n)) = w_1^{(n)}...w_j^{(n)}...w_{l_{n+1}}^{(n)}$$

The internal similarity between the fragments $w_j^{(n)}$ of $s^{(n)}$ is then calculated. For that purpose, a classification operation $cls(s^{(n)}, cc(n))$ is defined, which takes as arguments the segmented representation of level n $s^{(n)}$ and a set of classification criteria cc depending on the level n. A classification criterion can be a similarity threshold, and the operation of similarity (Euclidean distance, cosine distance, etc.) to apply to the spectrum, descriptor, or symbolic string. Thus, the *level n+1 representation* is the word $r^{(n+1)}$ obtained as follows

$$\begin{aligned} r^{(n+1)} &= cls(s^{(n)}, cc(n)) = cls(sgm(r^{(n)}, cs(n)), cc(n)) \\ &= x_1^{(n+1)}...x_j^{(n+1)}...x_{l_{n+1}}^{(n+1)} \end{aligned}$$

where the $x_j^{(n+1)}$ are letters defined on a new ordered alphabet $\mathcal{A}^{(n+1)}$.

3.2 Real-time formalisation

The formal diagrams are, in fact, not obtained in a single block, but in real-time, i.e., as the signal is acquired window after window in MORFOS and the initial representation is parsed in the MORFOS Algorithm.

Thus, a multi-scale formal diagram is defined by a sequence of states represented by the word $R = X_0.X_1...X_n$ with n the size of the word corresponding to the number of formal diagrams created. It is obtained by the concatenation of the triplets $X_j = (r^{(j)}, s^{(j)}, k^{(j)})$, where $r^{(j)}$ is the representation of level j, and $s^{(j)}$ is the segmented representation of level j. A marker of level j, $k^{(j)}$ is also defined, corresponding to the index of the letter where the last segmentation took place.

First, $R = \epsilon$ is initialised with the empty word. There are four operations that allow to change the state of R:

- the *creation of a new level* $X_n = (\epsilon, \epsilon, 0)$ with the operation nlc,
- the *real-time acquisition* at level 0 noted rta,
- the *real-time segmentation* at level n $rts(n)$,
- the *real-time classification* at level n $rtc(n)$ towards level $n + 1$

Not only do the $rts(n)$ segmentation and $rtc(n)$ classification operations apply at level n, but they can also take into account criteria depending on the level to which they apply. As a reminder, the real-time classification $rtc(n)$ occurs between levels n and $n + 1$ as it classifies the succession of objects of level n into a single object of level $n + 1$.

We also use the notations "." for the concatenation between two words or between a word and a letter, and $[r^{(j)}]_{k_1,k_2}$ for the subword of $r^{(j)}$ going from index k_1 to index k_2, included.

When acquiring a new object x at level 0, we concatenate $r^{(0)}$ with the new object x which comes from the flow:

$$\begin{aligned} X_0X_1...X_n &= (r^{(0)}, s^{(0)}, k^{(0)})X_1...X_n \\ &\xrightarrow[x]{rta} (r^{(0)}.x, s^{(0)}, k^{(0)}).X_1..X_n \end{aligned}$$

After the acquisition, a segmentation operation can be carried out if relevant.

During segmentation, we concatenate $s^{(j)}$ with the fragment composed of all letters acquired since the last segmentation (indicated by the index corresponding to the letter $k^{(j)}$) up to the previously acquired letter of index $l - 1$ where l is the length of the word of the representation $r^{(j)}$:

$$\begin{aligned} &X_0...(r^{(j)}, s^{(j)}, k^{(j)})...X_n \\ \xrightarrow{rts(j)} &X_0...(r^{(j)}, s^{(j)}.([r^{(j)}]_{k^{(j)},l-1}), l)...X_n \end{aligned}$$

After each segmentation, we apply a classification operation ctr on the last factor concatenated to $s^{(j)}$. We then concatenate the corresponding label obtained in $\mathcal{A}^{(j)}$ to $r^{(j+1)}$. If we denote w as the last fragment of our segmented representation, we then have:

$$\begin{aligned} &X_0...(r^{(j)}, s^{(j)}.w, k^{(j)})(r^{(j+1)}, s^{(j+1)}, k^{(j+1)})...X_n \\ \xrightarrow{rtc(j)} &X_0...(r^{(j)}, s^{(j)}.w, k^{(j)})(r^{(j+1)}.rtc(w), s^{(j+1)}, k^{(j+1)})...X_n \end{aligned}$$

We see that the acquisition operation rta is in reality a special case of the classification operation $cls(n)$ at level 0 with the classification function being the identity function.

It is possible that level $j + 1$ does not yet exist. Before proceeding with the classification operation, it is therefore necessary to create a new level with the operation nlc, which corresponds to the concatenation of the formal multi-scale diagram of size $n - 1$ with the formal diagram $X_n = (\epsilon, \epsilon, 0)$.

$$X_0X_1...X_{n-1} \xrightarrow{nlc} X_0X_1...X_{n-1}.(\epsilon, \epsilon, 0)$$

The final state of the word is obtained when all of the information at level 0 is acquired and the classification has been carried out at the highest level.

Thus, the set of n formal diagrams that constitute the multi-scale formal diagram is the ${r^{(j)}}_{j\in[0,n]}$ after having carried out the entirety of the changes of state.

The order of operations is shown in the diagram in Figure 3. The algorithm always starts with a succession of acquisitions at level 0, before segmenting. The segmentation is always followed by a classification, with an intermediate creation of a new level if it does not already exist. After classification, there is either segmentation at the higher level or acquisition at level 0.

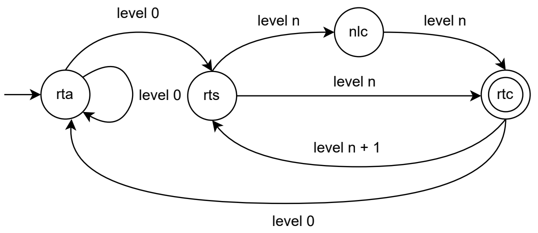

Figure 3. MORFOS Algorithm operations automaton. Each state corresponds to an operation, and the arcs correspond to the level of state affected by the operation to which the arrow points.

3.3 Example of W.A. Mozart's *Rondo* K.545

Thus, on the example of W.A. Mozart's *Rondo* presented in Figure 2, we have the following changes of state, according to the given segmentation and classification rules.

The multi-scale formal diagram is initially represented by the empty word ϵ. We want to acquire a new item at level 0. To do this, level 0 must first be created with the nlc operation:

$$\epsilon \xrightarrow{nlc} X_0$$

A new object a is then acquired at level 0:

$$(r^{(0)}, s^{(0)}, k^{(0)}) = (\epsilon, \epsilon, 0)$$

$$\xrightarrow[a]{rta} (r^{(0)}.a, s^{(0)}, k^{(0)}) = (a, \epsilon, 0)$$

There is no need to segment, so the next object, b, is acquired:

$$(a, \epsilon, 0) \xrightarrow[b]{rta} (ab, \epsilon, 0)$$

In the same way, let us suppose that the algorithm continues to acquire the next object, a:

$$(ab, \epsilon, 0) \xrightarrow[a]{rta} (aba, \epsilon, 0)$$

Here, we suppose that a segmentation operation is performed between b and a:

$$(aba, \epsilon, 0) \xrightarrow{rts(0)} (aba, (ab), 3)$$

In order to proceed with the classification of (ab) at the higher level, we must first initialise the higher level:

$$(aba, (ab), 3) \xrightarrow{nlc} (aba, (ab), 3).(\epsilon, \epsilon, 0)$$

The classification operation is then carried out on (ab), which is concatenated to $r^{(1)}$:

$$(aba, (ab), 3).(\epsilon, \epsilon, 0) \xrightarrow{rtc(0)} (aba, (ab), 3).(\text{A}, \epsilon, 0)$$

where A is the label given to the higher-level object.

And so on until the acquisition of all materials of level 0.

4. IMPLEMENTATION

In this section, we describe the implementation of the two specific operations of segmentation *rts* and classification *rtc*.

4.1 The data structures

In this subsection, we present three data structures necessary to understand the following algorithms.

- The *ongoing concatenated object of level n* corresponds to the concatenation of the objects acquired since the last segmentation.
- The *oracle of level n* is based on the *Variable Markov Oracle*, which is an automaton presented by Allauzen et al. [23] for compression and used for music improvisation by G. Assayag, C. Wang and S. Dubnov [24, 25]. It quickly finds the longest string of similar characters previously heard in the software. Each internal link to a state represents an object, and each state corresponds to the observation at time t of a discrete time series corresponding to the analysed stream.
- The *Material Memory of level n* constists of the *History Table of level n*, which contains the pairs of objects of level n and their corresponding objects of level n-1, and the *self-similarity matrix of level n*, which contains the similarity value between all acquired materials of level n.

These data structures are more precisely documented on [5].

4.2 The segmentation algorithm

The segmentation in MORFOS is performed by the function `segmentation(level n, object c, criteria cs): boolean` which returns 1 if there is segmentation and 0 otherwise. The level n provides access to all the data structures of the current level n and the c object provides information about the last object acquired.

The segmentation criteria correspond to a set of rules selected before the analysis and used to determine whether or not segmentation exists. These criteria are applied either to the signal, directly to the spectrum, or to descriptors (MFCC) or to symbols (MIDI or character strings). From a signal perspective, segmentation can be performed based on the dissimilarity computed either on frequency or dynamics. From a symbolic perspective, grammar rules are implemented by analysing changes in sequences of multiple same characters, the return to the beginning of a sequence, or length of sequences, to name just a few rules among many others.

The segmentation rules are designed in a modular way so that developers can add new ones.

4.2.1 Implementation of segmentation rules

Segmentation rules can be of three types: *mandatory*, *prohibited*, or *bivalent*. Mandatory segmentation rules necessarily imply segmentation at the moment preceding the acquisition of the current object. They therefore return the value 1 if there is segmentation between the acquired object x_{i+1} and the last object x_i. For example, `rule_1_similarity_word(ms_oracle,` `level)` is a mandatory segmentation rule that enforces the segmentation if the ongoing concatenated object of level *level* is considered similar to a previously segmented factor of the same level.

Prohibited segmentation rules forbid segmentation between the currently acquired object and the previous one. They therefore return the value 1 if there is no segmentation between the acquired object x_{i+1} and the last object x_i. For example, `rule_2_validated_hypothesis(ms_oracle, level)` is a prohibited segmentation rule that forbids segmentation if the ongoing concatenated object of level *level* is the beginning of a previously segmented factor of the same level.

Bivalent segmentation rules induce segmentation prohibition constraints and segmentation obligation constraints. These are implemented independently in two rules: mandatory segmentation and prohibited segmentation rules. For example `rule_6a_low_bound(ms_oracle, level)` and `rule_6b_high_bound(ms_oracle, level)` are respectively the prohibited segmentation and mandatory segmentation rules of the bivalent segmentation rule that forbid the segmentation if the ongoing concatenated object of level *level* has a length inferior to a predefined integer, and enforce a segmentation if the ongoing concatenated object of level *level* is of length superior to another predefined integer

When the user selects the rules for the analysis, they are added to the *list of rules* called by the rules manager.

4.2.2 Rules manager

The rules manager calculates the Boolean values associated with the result of the execution of each of the segmentation rules in the given context. This produces a set of booleans $\{r_i\}_{1\leq i\leq m_1}$ where m_1 is the number of rules selected for the analysis, with each bivalent rule being divided into two rules.

To determine whether segmentation is appropriate, the following logical formula is calculated:

$$b = \Big(\bigvee_{r_i\in\mathcal{O}\cap\mathcal{S}} r_i\Big) \wedge \Big(\bigwedge_{r_j\in\mathcal{I}\cap\mathcal{S}} \neg r_j\Big)$$

Where $\mathcal{O}$ is the set of mandatory rules, $\mathcal{I}$ is the set of forbidden rules, $\mathcal{S}$ is the set of selected rules, and b is the boolean result of the logical formula.

4.3 The classification algorithm

The classification in MORFOS is performed by the function `classification(level n, criteria cs): material`. This function adds the ongoing concatenated object at the current level to the higher-level oracle. To add this object to the oracle, it is compared in an optimised manner according to the construction of the oracle.

The classification criteria correspond to a set of rules selected before the analysis and are applied either to the signal or to symbols. From a signal perspective, classification is performed by computing, for example, cosine similarity or Euclidean distance on the spectrum or descriptors. From a symbolic perspective, strict equality or Needleman-Wunsch alignments [26] are computed on sequences.

The algorithm 1 is used for this purpose. To find the material corresponding to the newly acquired object, the ongoing concatenated object of level n is compared with the factor of level n associated with the previous objects of level $n+1$, accessed by the construction of the oracle of level $n+1$, and accessible from the history table in the material memory of level n. The similarity calculation `similarity(`w_i`,` $o_c^{(n)}$`, cc)` is applied to the corresponding factor and the object being formed at level n. Furthermore, let us assume that the material c obtained is new. In that case, the self-similarity matrix of the level n is updated with all the similarity values calculated previously and stored during the call to the $similarity$ function. Thanks to the construction of the oracle, all existing materials are compared when a new one is created.

Algorithm 1 classification(level n, criteria cs): object

```
For $x_i$ accessed in $oracle^{(n+1)}$ by going back and retrieving the suffix links starting from the last state:
    $w_i \leftarrow Th^{(n)}(x_i)$
    $s \leftarrow similarity(w_i, o_c^{(n)}, cc)$
    If $s = 1$:
        $c \leftarrow x_i$
        end For
    Else:
        $c \leftarrow create\_material()$
        $update(S^{(n)})$
return $c$
```

Finally, the function `similarity(factor` f_1`, factor` f_2`, classification criteria cc):boolean` calculates the similarity between two factors f_1 and f_2 based on classification criteria cc which are the classification rules, corresponding to similarity measures, selected by the user. Then, the algorithm checks if the value obtained exceeds a similarity threshold ss, also defined as a parameter. This amounts to taking the integer part greater than the difference between the similarity value and the threshold.

$$similarity(f_1, f_2, cc) = \lceil \frac{\sum_{sim_k\in cc} sim_k(f_1, f_2)}{Card(cc)} - ss \rceil$$

5. SECOND-ORDER ANALYSIS AND ATTENTION PHENOMENA

In this section, we analyse the algorithm's behaviour in order to better understand the phenomena of attention when listening to music. To do this, we rely on the notion of *cognitive phases* presented by J.M. Chouvel in [4].

5.1 Cognitive phases

Cognitive phases correspond to the characterisation of behaviours related to the handling of the information flow in the Cognitive Algorithm, i.e., the audio or symbolic flow during music listening.

These cognitive phases are linked to the results of the two main tests of the Cognitive Algorithm as originally presented, namely the paradigmatic test and the syntagmatic test.

These phases make it possible to clarify specific cognitive states:

Integration: Integration occurs when there is a sensation of continuity; when the current object is concatenated with the objects heard since the last segmentation, and a higher-level object is consequently under construction.

Retention: J.M. Chouvel writes in [27] that *"The retentional memory required by form must allow the memorisation of elementary events"*. Thus, retention specifically refers to the memorisation of information in the most compressed form possible, with minimal information loss. It involves memorising each new material. Retention occurs when the classification test fails (a new material is analysed).

Realisation: In the algorithm, realisation occurs when an object has been recognised at the current level and allows a hypothesis that the whole corresponding object at the superior level will occur. The following object at the current level is then expected by the perception. Realisation occurs when the classification test is valid.

Acquisition/Recursion: Acquisition refers to the grouping phenomenon of recently heard objects to form a higher-level object, with the aim of processing this new object as a whole. Recursion is the invocation of the Cognitive Algorithm at the higher level on this newly acquired object. There is therefore acquisition at the higher level and recursion at the algorithm level when *completion* occurs, meaning the segmentation test is valid.

Expectation of the Unknown: Expectation of the unknown arises when no hypothesis can be made about the *successor object*. The successor object refers to the object acquired at the current level after the object currently being acquired. Otherwise, the listener is in the *expectation of the known*. However, even if the known is expected, the actually acquired object may not correspond to the expected one. This gives rise to possibilities of *satisfied prediction* or *unsatisfied prediction*. A satisfied prediction occurs when the subsequently acquired object matches the emitted hypothesis, and an unsatisfied prediction otherwise.

In MORFOS, the classification and segmentation tests are performed in reverse order compared to Chouvel's model. We consider segmentation as prior to classification: before classifying an element, it must first be distinguished. At the algorithmic level, the software first performs automated signal sampling, or preliminary segmentation, that provides a sequential symbolic representation. In reality, these two tests likely occur simultaneously: the algorithm's implementation shows that classification immediately follows segmentation, but similarity computations are performed beforehand to determine segmentation. This deserves further discussion.

Thus, the modeling of cognitive phases in our software is presented in Figure 4. We introduce the term *object currently being acquired* x_t^j to denote the object currently acquired at level j and time t. It should not be confused with the *ongoing concatenated object*, which corresponds to the concatenation of objects at a given level in order to build a higher-level object. The *successor object* is the object that immediately follows the currently acquired object at the same level, i.e., the object x_{t+1}^j acquired at time $t+1$ at level j. We may also extend this terminology to refer to the *higher-level acquired object* x_k^{j+1}, which is the higher-level object closed by x_t^j, and the *higher-level successor object* x_{k+1}^{j+1}, which is the higher-level object initiated by x_{t+1}^j.

An *iteration* of the algorithm involves acquiring an object at a given time t and level j and applying the segmentation and classification tests to it. Two iterations are shown in Figure 4. The first (top) corresponds to the *segmentation/classification* iteration at level $j-1$ giving the object x_t^j, while the second (bottom) corresponds to the *segmentation/classification* iteration during the acquisition of object x_{t+1}^j.

The two tests relevant for determining the cognitive phase are highlighted in dark gray.

A major difference between our algorithm and the one proposed by J.M. Chouvel is that the segmentation test validates or invalidates a segmentation between the currently acquired object and the previous one. It does not indicate that the current object *closes* the acquired higher-level object, but rather that the successor object *initiates* the successor higher-level object.

The first is the classification test, which corresponds to the classification of the factor $w_t^{(j-1)}$ at level $j-1$: it labels this factor into a current acquisition object $x_t^{(j)}$ at level j. The segmented object that yields this factor is not our focus here. If this label matches that of a material seen previously, the classification test is considered *valid*. Otherwise, the test is *invalid*.

The second test is the segmentation test, which determines whether the object $x_t^{(j)}$ constitutes a closing element. This means that during the acquisition of its successor $x_{t+1}^{(j)}$, segmentation occurs into a factor $w_k^{(j)} = (x_{t-i}^{(j)}...x_t^{(j)})$. If segmentation occurs, the test is said to be *valid*. Otherwise, when there is no segmentation, the segmentation test is *invalid*.

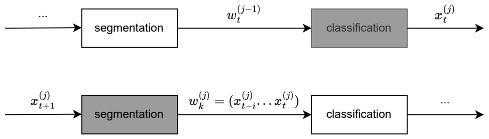

Figure 4. Modeling of cognitive phases.

We now present the four cognitive phases.

5.1.1 Cognitive Phase 1

Cognitive phase 1 corresponds to an invalid classification test and an invalid segmentation test. An invalid classification test at level j means that the currently acquired object x_t^j is not similar to any previously heard object at level j:

thus, the corresponding material is memorised. Furthermore, there is *integration* in the sense that the object is concatenated with the objects heard since the last segmentation (construction of the factor $w_k^j = x_{t-i}^j ... x_t^j$, where $t - i > 0$ is the index of the last segmentation), and the listener is in the process of constructing a higher-level object. However, during the acquisition of x_{t+1}^j, there is no segmentation: x_t^j does not close the higher-level object under construction, and nothing happens at the upper level.

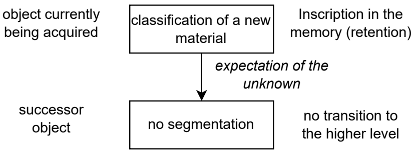
object currently being acquired
classification of a new material
Inscription in the memory (retention)
expectation of the unknown
successor object
no segmentation
no transition to the higher level

Figure 5. Phase 1.

In terms of hypotheses: since the object $x_t^{(j)}$ corresponds to new material acquired at the current level, no specific hypothesis is made regarding the object to follow $x_{t+1}^{(j)}$, nor for the higher-level object $x_k^{(j+1)}$. More precisely, a novelty hypothesis is put forward: the listener expects the unknown.

5.1.2 Cognitive Phase 2

Cognitive phase 2 corresponds to an invalid classification test and a valid segmentation. Thus, memorisation and retention occur in the memory of the currently acquired object x_t^j as in phase 1. However, there is now segmentation before x_{t+1}^j during its acquisition, which means that the object under construction $w_k^j = x_{t-i}^j ... x_t^j$ is acquired and therefore labeled (through a classification operation) as a higher-level object, with a recursive call to the MORFOS algorithm.

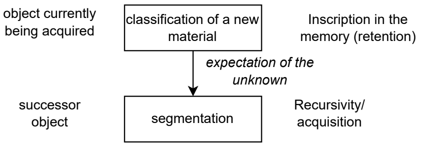
object currently being acquired
classification of a new material
Inscription in the memory (retention)
expectation of the unknown
successor object
segmentation
Recursivity/ acquisition

Figure 6. Phase 2.

As in cognitive phase 1, new material is acquired at the current level, so that a hypothesis of novelty arises concerning the object to come. Moreover, the system acquires probably new material at the higher level (unless this is considered a negligible variation) for the object $x_k^{(j+1)}$, and a novelty hypothesis arises concerning the following object $x_{k+1}^{(j+1)}$. The listening subject therefore expects the unknown, both at the current level and at the higher level.

5.1.3 Cognitive Phase 3

Cognitive phase 3 corresponds to a valid classification test and an invalid segmentation. The currently acquired object x_t^j is similar to a previously heard object. The associated time instant t is memorised, but the corresponding material is not memorised again. The object is concatenated to the ongoing concatenated object, and there is a *realisation* phase: potential materials are concretised in memory. As in phase 1, however, there is no segmentation after acquiring the successor object x_{t+1}^j.

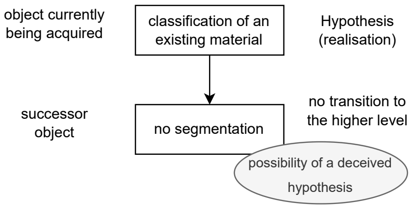
object currently being acquired
classification of an existing material
Hypothesis (realisation)
successor object
no segmentation
no transition to the higher level
possibility of a deceived hypothesis

Figure 7. Phase 3.

Recognition in x_t^j of already perceived material allows hypotheses to be made concerning the successor object x_{t+1}^j expected at the current level. Hypotheses are based on objects that succeeded previous occurrences of the material corresponding to x_t^j. This is also made possible by the fact that there is no segmentation after $x_t^{(j)}$. Indeed, if there were segmentation after $x_t^{(j)}$, then the object x_{t+1}^j would correspond to the beginning of a new object at level j, and no hypothesis would be made (case of phase 4). Here, hypotheses are made: the listening subject expects the known. However, it is possible that the object x_{t+1}^j does not correspond to the expected materials: a surprise phenomenon will then occur.

In our modeling, it is the structure of the Variable Markov Oracle that allows us to find previous occurrences and their successors, which are potential candidates for $x_{t+1}^{(j)}$.

It should be noted that at this stage, hypotheses can also be made about the higher-level object $x_k^{(j+1)}$: either the object under construction w_k^j is the prefix of one or more higher-level objects already seen, and these objects are hypotheses of realisation, or the object currently being constructed does not correspond to any prefix and the unknown is therefore expected at the higher level. By moving up the levels, the scope of anticipation can therefore potentially be increased.

5.1.4 Cognitive Phase 4

Cognitive phase 4 corresponds to a valid classification test (realisation) and also a valid segmentation test (acquisition at the higher level and recursive call of the algorithm).

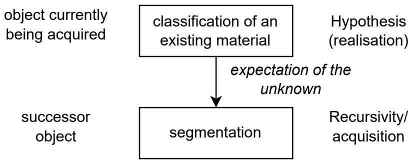
object currently being acquired
classification of an existing material
Hypothesis (realisation)
expectation of the unknown
successor object
segmentation
Recursivity/ acquisition

Figure 8. Phase 4.

As in phase 3, acquiring a recognised object x_t^j allows hypotheses to be made about the object $x_{t+1}^{(j)}$, but in reality, there is segmentation between these two objects. The object $x_{t+1}^{(j)}$ is thus the beginning of a new factor w_{k+1}^{j} and no hypothesis is made at level j: the unknown is expected.

At the higher level, however, it is possible that the obtained object x_k^{j+1} corresponds to already heard material at level $j+1$: thus, a hypothesis can be made about the successor object x_{k+1}^{j+1}.

5.2 Analyses

The computation of the phases in real time has been implemented in MORFOS. We computed them for different types of segmentation on Mozart's *Rondo*:

- no segmentation;
- factors of size two are segmented;
- random segmentation;
- set of relevant segmentation rules;

The corresponding multi-scale formal diagrams are represented respectively in Figures 9, 10, 11, and 12.

5.2.1 No Segmentation

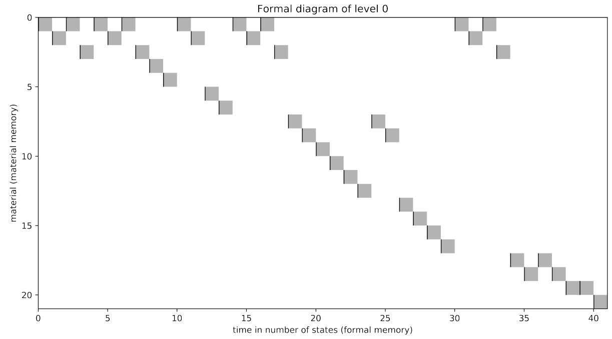

Figure 9. Automated (Multi-scale) formal diagram of W.A. Mozart's *Rondo K545* obtained with MORFOS with no segmentation. As there is no segmentation, there is only one level of hierarchy.

Table 1 shows the phases corresponding to an analysis without segmentation as shown in the multi-scale formal diagram Figure 9. The first row corresponds to the states acquired at level 0, and each additional row corresponds to a level and contains the phases corresponding to the acquisition of the states associated with the segmentation of the state at level 0. In this case, we observe only phases 1 and 3: phase 1 appears during the acquisition of a new material and phase 3 during the return to an already seen material, but there is never any segmentation except for the last material, which is considered segmented by default (and was never seen, which yields phase 2).

There are only phases 1 and 3, so there is alternation of retention and realisation in the memory. Moreover, there is no transition to the higher-level, so the number of concatenated objects in the ongoing concatenated object of level 0 keeps increasing. Therefore, a hypothesis is that it increases the cognitive load as no compression of the information into higher level segments is made.

5.2.2 Segmentation every 2 objects

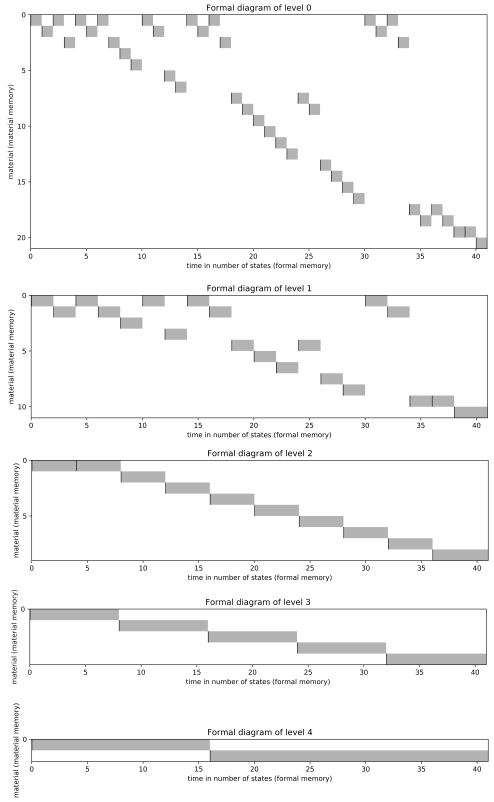

Figure 10. Automated Multi-scale formal diagram of W.A. Mozart's *Rondo K545* obtained with MORFOS, where factors of size two are segmented.

Table 2 presents the cognitive phases obtained during an analysis where size-two factors are segmented. Segmenting every two objects presents the cognitive phase 1 or 3 every one time, and phase 2 or 4 every other time, depending on whether the acquired object was already in memory or not.

Starting from level 2, there are almost only phases 1 and 2 due to the classification of objects from materials that have never been seen before; therefore, there is constantly retention at the acquisition of objects at levels 2 and 3. Nevertheless, as new materials constantly appear, there is always an expectation of the unknown.

5.2.3 Random segmentation

We present in the table 3 the cognitive phases associated with a random segmentation as shown in the multi-scale formal diagram Figure 11. The phases obtained also seem random between 1 and 3 when there is no segmentation (depending on whether the object has already been acquired or not) and between 2 and 4 if there is random segmentation.

Random segmentation favors the non-recognition of the objects starting at level 1, so there are only phases 1 and 2 at level 1 and higher levels. There is thus only retention in the memory except for level 0.

level	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u
0	1	1	3	1	3	3	3	3	1	1	3	3	1	1	3	3	3	3	1	1	1	1	1	1	3	3	1	1	1	1	3	3	3	3	1	1	3	3	1	3	2

Table 1. Cognitive phases computed at each level of segmentation (here, only level 0 is computed) according to W.A. Mozart's *Rondo K545* multi-scale formal diagram obtained with MORFOS with no segmentation.

level	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u
0	1	2	3	2	3	4	3	4	1	2	3	4	1	2	3	4	3	4	1	2	1	2	1	2	3	4	1	2	1	2	3	4	3	4	1	2	3	4	1	3	2
1		1		2		3		4		1		4		1		4		3		2		1		2		3		2		1		4		3		2		3			2
2				1				4				1				2				1				2				1				2				1					2
3								1								2								1								1									2
4																1																									2

Table 2. Cognitive phases computed at each level of segmentation according to W.A. Mozart's *Rondo K545* multi-scale formal diagram obtained with MORFOS, where factors of size two are segmented.

level	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u
0	1	2	3	1	3	3	3	4	1	1	3	3	2	1	3	4	3	4	1	2	1	2	1	1	3	4	1	2	1	1	4	3	3	3	2	1	4	3	2	3	2
1		1						2					1			2		1		2		1				2		1			1				2		1		1		2
2								1								2				1						1									1						2
3																1																									2

Table 3. Cognitive phases computed at each level of segmentation according to W.A. Mozart's *Rondo K545* multi-scale formal diagram obtained with MORFOS with random segmentation.

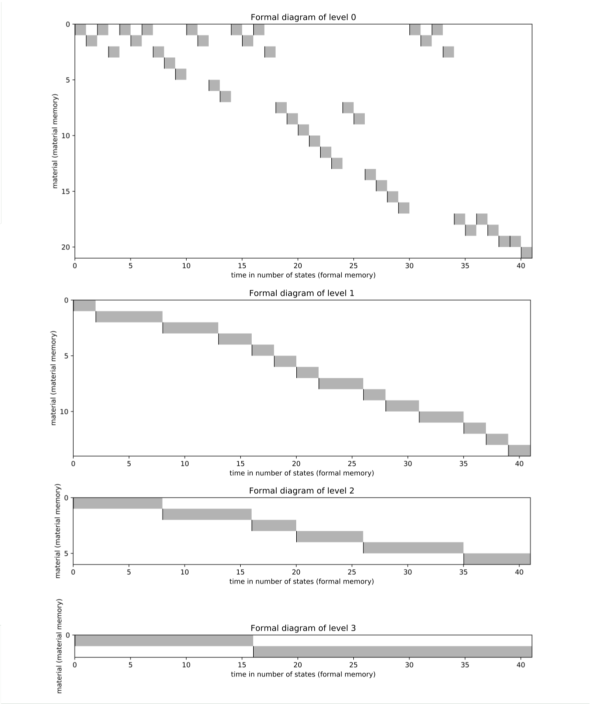

Figure 11. Automated Multi-scale formal diagram of W.A. Mozart's *Rondo K545* obtained with MORFOS with random segmentation.

5.2.4 *Set of relevant segmentation rules for the agenda*

Table 4 presents the phases corresponding to a set of rules as defined as follows :

- segmentation when the beginning of a higher-level object, i.e., the start of a factor in the current-level segmented representation, is found.
- As soon as the ongoing concatenated object in the current level is an object already seen in the higher level, the character string is segmented.
- If the ongoing concatenated object at the current level is a prefix of a factor in the segmented representation at the current level, but the object acquired is also identical to the object following the prefix, then the string is not segmented.

The corresponding multi-scale formal diagram obtained is shown in Figure 12.

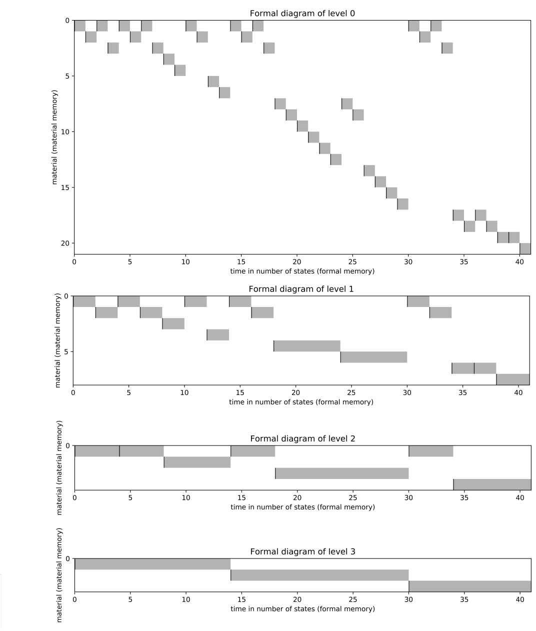

Figure 12. Automated Multi-scale formal diagram of W.A. Mozart's *Rondo K545* obtained with MORFOS with a set of relevant segmentation rules described in the article.

level	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u
0	1	2	3	2	3	4	3	4	1	2	3	4	1	2	3	4	3	4	1	1	1	1	1	2	3	3	1	1	1	2	3	4	3	4	1	2	3	4	1	3	2
1		1		2		3		4		1		3		2		3		4						1						2		3		4		1		3			2
2				1				3						2				3												2				3							2
3														1																1											2

Table 4. Cognitive phases computed at each level of segmentation according to W.A. Mozart's *Rondo K545* multi-scale formal diagram obtained with MORFOS with a set of relevant segmentation rules described in the article.

This example can be compared to the one presented by J.M. Chouvel in [27], which is recalled in Figure 13.

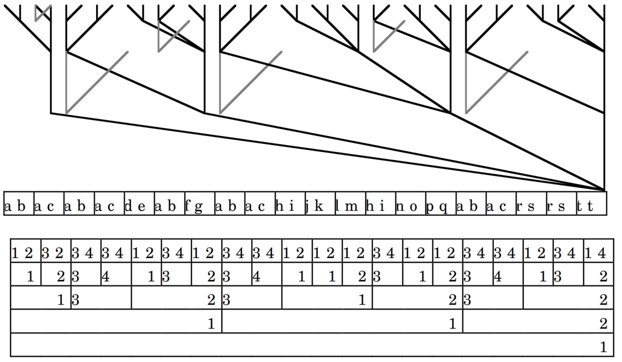

Figure 13. Cognitive phases of W.A. Mozart's *Rondo K545* according to Chouvel in [27].

This is an opportunity for us to introduce here another representation presented by Jean-Marc Chouvel of the work called the *structural representation*: here, the meaning of the branches, in addition to informing about the structure of the work, informs about the phenomenon of continuity of information (backward branching) or anticipation (forward branching). Forward branches that cross thus constitute disappointed anticipations.

We indeed obtain the same segmentations as those obtained by Chouvel, with a few exceptions. The first is the last character u, which is considered included in the last t of Chouvel's analysis: whereas for him it is a repeated and segmenting character (phase 4), we divide it into two characters t and u, the first being repeated and non-segmenting (phase 3) and the second being new and segmenting (phase 2).

The second difference concerns the unfolding of the two developments $(hijklm)$ and $(hinopq)$. Whereas Chouvel notes phase 2 at the acquisition of the first i, k, and o, we here consider phase 1 for new objects without segmentation, and phase 3 for the second i (object already seen without segmentation), where he considers phase 4 because there is segmentation. The objects segmented at level 0 are then propagated to the higher level 1 into three objects, which themselves have no impact on level 2.

6. DISCUSSION

A high-level notation, the formal diagram, is presented in this article. This notation is important for musicians as it might help them memorising the musical structure, and it has been used in teaching methods for memorise large forms. The relevance of the software has been more or less demonstrated using an educational example to facilitate understanding and validation of the concepts, although the aim of the software is to explore a larger repertoire.

We might wonder to what extent classification and segmentation criteria correspond to the associated cognitive process while listening to music. In this context, we could compare the Cognitive Algorithm with the cognitive model of music processing presented by S. Koelsch et W. A. Siebel [28]. Thus, the potentials that seem most relevant are those detecting attention and the end of phrases (MMN, ERAN, CPS). However, the musical context is of prime importance in pattern recognition, so in order to observe these potentials, it is necessary to carry out studies on specially composed music that allows all factors, such as transposition, to be controlled during pattern recognition, which makes such a study difficult to carry out.

To better understand the cognitive phases, we could also compare the computational costs associated with the acquisition of each musical object with the associated cognitive phases. We could compute the complexity of each object according to Kolmogorov's definition [29], meaning that we count how many operations are necessary to describe an object according to its context, i.e., the previous objects. Are the computational costs higher when a hypothesis is deceived after the cognitive phase 3? To what extent are the cognitive phases correlated with the compression of the information? These questions need to be more thoroughly studied.

7. CONCLUSION

In this article, the formalisation and implementation of classification and segmentation tests of the Cognitive Algorithm within the framework of a cognitive listening model with MORFOS software is presented. Based on this model, the cognitive processes associated with segmentation or not, and with the classification of a known or unknown element following the analysed musical object, have been implemented, giving four processes called cognitive phases. An intuition of the actual cognitive phenomena associated with cognitive phases can be given through the notion of integration, retention, realisation, acquisition/recursion, and expectation of the unknown. Then, four analyses of these cognitive phases are presented on four distinct segmentation examples of the *Rondo* from the *Sonata K.545* by W.A. Mozart.

In a second stage, an idea would be to calculate secondary cognitive phases, which can be defined as the *derivative* of cognitive phases. J.M. Chouvel [27] presents sixteen secondary cognitive phases, corresponding to the combinatorial succession of the four existing phases. By implementing them in MORFOS, we might try to better understand how cognitive processes succeed each other and their impact on the cognitive load.

Acknowledgments

We would like to thank the Studio for Creation and Research in Computer Science and Experimental Music (SCRIME) for their contribution to the funding of this article.

8. REFERENCES

[1] W. Fitch and M. Martins, “Hierarchical processing in music, language, and action: Lashley revisited,” pp. 2014 May;1316(1):87–104, 2014.

[2] F. Lerdhal and R. Jackendoff, *A generative theory of tonal music*. Cambridge, Mass. : MIT Press, 1983.

[3] S. Tojo, K. Hirata, and M. Hamanaka, “Computational reconstruction of cognitive music theory.” *New Gener. Comput.*, vol. 31, p. 89–113, 2013.

[4] J.-M. Chouvel, “Musical form, from a model of hearing to an analytic procedure,” *Interface*, vol. 22, pp. 99–117, 1993.

[5] J. Calandra, J.-M. Chouvel, and M. Desainte-Catherine, “Hierarchisation algorithm for morfos : a music analysis software,” in *2025 International Computer Music Conference (ICMC 2025) Boston, MA, United States.*, Jun 2025.

[6] N. Meeùs, “Épistémologie d’une musicologie analytique,” vol. volume XXII, pp. 97–114, 2015.

[7] E. Bigand and S. McAdams, *Contributions de la musique aux recherches sur la cognition auditive humaine*, 2005, ch. 8, pp. 261–262.

[8] X. Sun, Q. He, Y. Gao, and W. Li, “Musical tempo estimation using a multi-scale network,” in *in Proc. of the 22nd Int. Society for Music Information Retrieval Conf., Online*, 2021.

[9] S. Böck, M. E. P. Davies, and P. Knees, “Multi-task learning of tempo and beat: Learning one to improve the other,” in *Int. Society for Music Information Retrieval Conference*, 2019.

[10] S. Samson, *Perception des timbres musicaux*, de boeck supérieur ed. Bernard Lechevalier éd., 2010, pp. 123–146.

[11] G. Gabbolini and D. Bridge, “An interpretable music similarity measure based on path interestingness,” in *in Proc. of the 22nd Int. Society for Music Information Retrieval Conf., Online*, 2021.

[12] D. Herremans, D. Martens, and K. Sörensen, *Composer Classification Models for Music-Theory Building*, 10 2015.

[13] M. Sarfati, A. Hu, and J. Donier, “Ensemble-based cover song detection,” 05 2019.

[14] W. De Haas, M. Robine, P. Hanna, R. Veltkamp, and F. Wiering, “Comparing approaches to the similarity of musical chord sequences,” vol. 6684, 06 2010, pp. 242–258.

[15] J. Foote, “Visualizing music and audio using self-similarity,” in *Proc. Of ACM Multimedia*, 1999, pp. 77–80.

[16] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, Chicago,, 1990.

[17] ——, *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. University of Chicago Press, Chicago,, 1992.

[18] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent, “System & contrast : a polymorphous model of the inner organization of structural segments within music pieces.” *Music Perception, University Of California Press*, vol. 33, pp. 631–661, 2016.

[19] O. Laske, *Music, Memory and Thought, explorations in Cognitive Musicology*. University Microfilms International, Ann Arbor (MI), 1977.

[20] J. Nattiez, *Musicologie générale et sémiologie*. Christian Bourgois, 1987.

[21] F. Delalande, *La musique au-delà des notes*. Presse Universitaires de Rennes, 2019.

[22] J. Calandra, J.-M. Chouvel, and M. Desainte-Catherine, “Multi-scale oracle and automated representation of formal diagrams based on the cognitive algorithm,” in *Int. Conf. On Technologies For Music Notation And Representation - TENOR 2021*, 2021.

[23] C. Allauzen, M. Crochemore, and M. Raffinot, “Factor oracle: a new structure for pattern matching; ; theory and practice of informatics.” in *Proceedings of SOFSEM’99*, Paris, France, 1999.

[24] G. Assayag and S. Dubnov, “Using factor oracles for machine improvisation,” *Soft Computing*, vol. 8, pp. 604–610, 01 2004.

[25] C. Wang and S. Dubnov, “The variable markov oracle: Algorithms for human gesture applications,” in *IEEE Multimedia, vol. 22, no. 4*, 2015, pp. 52–67.

[26] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[27] J.-M. Chouvel, *Esquisses pour une pensée musicale, Les métamorphoses d’Orphée*. L’Harmattan, Paris, 1998.

[28] S. W. Koelsch S, “Towards a neural basis of music perception,” vol. Dec.9(12), pp. 97–114, 2005.

[29] A. N. Kolmogorov, “Three approaches to the quantitive definition of information,” *In : Prob Info Trans*, vol. 1.1, p. 3–11, 1965.